WHAT IS CLAIMED IS:

1. A method for finding a next free bit in a register having N bits and a current pointer pointing to one of the bits, the method comprising:

obtaining the value P of the current pointer, wherein $0 \leq P \leq N-1$, and P and N are integers;

generating an add vector, with its bit number P set; and

adding the add vector to a check vector in the register to obtain a sum.

2. The method according to claim 1, further comprising:

multiplying the sum with an inverse of the check vector to find the next free bit.

3. An apparatus for performing the method according to claim 1, the apparatus comprising:

a calculator for obtaining the value P of the current pointer;

a generator for generating the add vector; and

an adder for adding the add vector to the check vector.

4. The apparatus according to claim 3, further comprising:

a multiplier for multiplying the sum with an inverse of the check vector to find the next free bit.

5. An apparatus for performing the method according to claim 1, the apparatus comprising:

means for obtaining the value P of the current pointer;

means for generating the add vector; and

means for adding the add vector to the check vector.

6. The apparatus according to claim 5, further comprising:

means for multiplying the sum with an inverse of the check vector to find the next free bit.

7. A method for finding a next free bit in a register having N bits and a current pointer pointing to one of the bits, the method comprising:

breaking the N bits of a check vector in the register into M parts, wherein N and M are integers and $1< M<N$; and

selecting an available part that has a free bit.

8. The method according to claim 7, wherein the available part is a part pointed to by the current pointer.

9. The method according to claim 8, further comprising:

finding a free bit in the available part.

10. The method according to claim 9, wherein the step for finding a free bit comprises:

breaking the current pointer into upper bits and lower bits, wherein the current pointer has X bits, the upper bits have Y bits and a value U, and the lower bits have X-Y bits and a value L, and wherein $0 \leq U \leq 2^{Y}-1$, and $0 \leq L \leq 2^{X-Y}-1$, where all of X, Y, U, and L are integers;

obtaining an add vector by setting its bit number L;

adding the add vector to the available part to obtain a sum; and

multiplying the sum with an inverse of the available part.

11. The method according to claim 7, wherein the available part is a first part, having a free bit, to the left of the part pointed to by the current pointer.

12. The method according to claim 11, wherein the step for selecting the available part comprises:

breaking the current pointer into upper bits and lower bits, wherein the current pointer has X bits, the upper bits have Y bits and a value U, and the lower bits have X-Y bits and a value L, and wherein $0 \leq U \leq 2^{Y}-1$, and $0 \leq L \leq 2^{X-Y}-1$, where all of X, Y, U, and L are integers;

creating a check sector, wherein each bit of the check sector results from performing an AND operation to all bits of a corresponding part of the M parts;

obtaining an add vector by setting its bit number U;

adding the add vector to the check sector to obtain a sum; and

multiplying the sum with an inverse of the check sector.

13. The method according to claim 11, further comprising finding a free bit in the available part.

14. The method according to claim 13, wherein the step for finding a free bit comprises:

increasing the available part by 1; and

multiplying the increased available part with an inverse of the available part.

15. The method according to claim 7, wherein the available part is a first part, having a free bit, from the beginning of the register.

16. The method according to claim 15, wherein the step for selecting the available part comprises:

creating a check sector, wherein each bit of the check sector results from performing an AND operation to all bits of a corresponding part of the M parts;

increasing the check sector by 1; and

multiplying the increased check sector with an inverse of the check sector.

17. The method according to claim 15, further comprising finding a free bit in the available part.

18. The method according to claim 17, wherein the step for finding a free bit comprises:

increasing the available part by 1; and

multiplying the increased available part with an inverse of the available part.

19. The method according to claim 7, further comprising:

creating a check sector, wherein each bit of the check sector results from performing an AND operation to all bits of a corresponding part of the M parts; and

deciding whether the register has a free bit by performing an AND operation to all bits of the check sector.

20. An apparatus for performing the method according to claim 7, the apparatus comprising:

a first breaker for breaking the N bits of the check vector; and

a selector for selecting the available part.

21. The apparatus according to claim 20, wherein the selector selects a part pointed to by the current pointer as the available part.

22. The apparatus according to claim 20, wherein the selector selects the available part on the left of the part pointed to by the current pointer.

23. The apparatus according to claim 22, further comprising:

a check sector generator for generating a check sector, wherein each bit of the check sector results from performing an AND operation to all bits of a corresponding part of the M parts.

24. The apparatus according to claim 23, further comprising:

a second breaker for breaking the current pointer into upper bits and lower bits, wherein the current pointer has X bits, the upper bits have Y bits and a value U, and the lower bits have X-Y bits and a value L, and wherein $0 \leq U \leq 2^{Y}-1$, and $0 \leq L \leq 2^{X-Y}-1$, where all of X, Y, U, and L are integers.

25. The apparatus according to claim 24, wherein the selector comprises:

an add vector generator, setting bit number U of the add vector;

an adder for adding the add vector to the check sector to obtain a sum; and

a multiplier for multiplying the sum with an inverse of the check sector.

26. The apparatus according to claim 20, wherein the selector selects the available part from the beginning of the register.

27. The apparatus according to claim 26, further comprising:

a check sector generator for generating a check sector, wherein each bit of the check sector results from performing an AND operation to all bits of a corresponding part of the M parts.

28. The apparatus according to claim 27, wherein the selector comprises:

an adder for increasing the check sector by 1; and

a multiplier for multiplying the increased check sector with an inverse of the check sector.

29. The apparatus according to claim 20, further comprising a free bit finder.

30. The apparatus according to claim 29, wherein the free bit finder finds a free bit on the left of the bit pointed to by the current pointer.

31. The apparatus according to claim 30, further comprising:

a second breaker for breaking the current pointer into upper bits and lower bits, wherein the current pointer has X bits, the upper bits have Y bits and a value U, and the lower bits have X-Y bits and a value L, and wherein $0 \leq U \leq 2^{Y}-1$, and $0 \leq L \leq 2^{X-Y}-1$, where all of X, Y, U, and L are integers.

32. The apparatus according to claim 31, wherein the free bit finder comprises:

an add vector generator, setting bit number L of the add vector;

an adder for adding the add vector to the available part to obtain a sum; and

a multiplier for multiplying the sum with an inverse of the available part.

33. The apparatus according to claim 29, wherein the free bit finder finds a free bit from the beginning of the available part.

34. The apparatus according to claim 33, wherein the free bit finder comprises:

an adder for increasing the available part by 1; and

a multiplier for multiplying the increased available part with an inverse of the available part.

35. The apparatus according to claim 20, further comprising:

a check sector generator for generating a check sector, wherein each bit of the check sector results from performing an AND operation to all bits of a corresponding part of the M parts; and

a register status unit for performing an AND operation to all bits of the check sector.

36. The apparatus according to claim 20, further comprising:

a next vector generator for generating the next vector with the found free bit masked.

37. An apparatus for performing the method according to claim 7, the apparatus comprising:

means for breaking the N bits of the check vector; and

means for selecting the available part.

38. The apparatus according to claim 37, wherein the selecting means selects the part pointed to by the current pointer as the available part.

39. The apparatus according to claim 37, wherein the selecting means selects the available part on the left of the part pointed to by the current pointer.

40. The apparatus according to claim 39, further comprising:

means for generating a check sector, wherein each bit of the check sector results from performing an AND operation to all bits of a corresponding part of the M parts.

41. The apparatus according to claim 40, further comprising:

means for breaking the current pointer into upper bits and lower bits, wherein the current pointer has X bits, the upper bits have Y bits and a value U, and the lower bits have X-Y bits and a value L, and wherein $0 \leq U \leq 2^{Y}-1$, and $0 \leq L \leq 2^{X-Y}-1$, where all of X, Y, U, and L are integers.

42. The apparatus according to claim 41, wherein the selecting means comprises:

means for generating an add vector, setting bit number U of the add vector;

means for adding the add vector to the check sector to obtain a sum; and

means for multiplying the sum with an inverse of the check sector.

43. The apparatus according to claim 37, wherein the selecting means selects the available part from the beginning of the register.

44. The apparatus according to claim 43, further comprising:

means for generating a check sector, wherein each bit of the check sector results from performing an AND operation to all bits of a corresponding part of the M parts.

45. The apparatus according to claim 44, wherein the selecting means comprising:

means for increasing the check sector by 1; and

means for multiplying the increased check sector with an inverse of the check sector.

46. The apparatus according to claim 37, further comprising:

means for finding the free bit.

47. The apparatus according to claim 46, wherein the free bit finding means finds the free bit on the left of the bit pointed to by the current pointer.

48. The apparatus according to claim 47, further comprising:

a second means for breaking the current pointer into upper bits and lower bits, wherein current pointer has X bits, the upper bits have Y bits and a value U, and the lower bits have X-Y bits and a value L, and wherein $0 \leq U \leq 2^{Y}-1$, and $0 \leq L \leq 2^{X-Y}-1$, where all of X, Y, U, and L are integers.

49. The apparatus according to claim 48, wherein the free bit finding means comprises:

means for generating an add vector, setting bit number L of the add vector;

means for adding the add vector to the available part to obtain a sum; and

means for multiplying the sum with an inverse of the available part.

50. The apparatus according to claim 46, wherein the free bit finding means finds the free bit from the beginning of the available part.

51. The apparatus according to claim 50, wherein the free bit finding means comprises:

means for increasing the available part by 1; and

means for multiplying the increased available part with an inverse of the available part.

52. The apparatus according to claim 37, further comprising:

means for generating a check sector, wherein each bit of the check sector results from performing an AND operation to all bits of a corresponding part of the M parts; and

means for performing an AND operation to all bits of the check sector.

53. The apparatus according to claim 37, further comprising:

means for generating the next vector with the found free bit masked.

54. A buffer management system comprising the apparatus according to claim 20, wherein the system allocates buffers to the register.

55. The buffer management system according to claim 54, further comprising:

an allocation state machine for controlling the apparatus; and

an allocation memory for providing a new line to the register.

56. The buffer management system according to claim 55, further comprising a buffer management state machine for controlling buffer allocation to the register.

57. The buffer management system according to claim 55, further comprising a clear arbiter for clearing the buffers allocated to the register.

58. The buffer management system according to claim 55, further comprising an allocation arbiter for arbitrating between at least two requests for buffers.

59. The buffer management system according to claim 55, further comprising an allocation counter for counting the allocated buffers.

60. The buffer management system according to claim 55, further comprising a reclaim mechanism.

61. A buffer management system comprising the apparatus according to claim 37, wherein the system allocates buffers to the register.

62. The buffer management system according to claim 61, further comprising:

means for controlling the apparatus; and

means for providing a new line to the register.

63. The buffer management system according to claim 62, further comprising means for controlling buffer allocation to the register.

64. The buffer management system according to claim 62, further comprising means for clearing the buffers allocated to the register.

65. The buffer management system according to claim 62, further comprising means for arbitrating between at least two requests for buffers.

66. The buffer management system according to claim 62, further comprising means for counting the allocated buffers.

67. The buffer management system according to claim 62, further comprising means for reclaiming buffers.

68. A computer software product containing program code for performing the method according to claim 1.

69. A computer software product containing program code for performing the method according to claim 2.

70. A computer software product containing program code for performing the method according to claim 7.

71. A computer software product containing program code for performing the method according to claim 8.

72. A computer software product containing program code for performing the method according to claim 9.

73. A computer software product containing program code for performing the method according to claim 10.

74. A computer software product containing program code for performing the method according to claim 11.

75. A computer software product containing program code for performing the method according to claim 12.

76. A computer software product containing program code for performing the method according to claim 13.

77. A computer software product containing program code for performing the method according to claim 14.

78. A computer software product containing program code for performing the method according to claim 15.

79. A computer software product containing program code for performing the method according to claim 16.

80. A computer software product containing program code for performing the method according to claim 17.

81. A computer software product containing program code for performing the method according to claim 18.

82. A computer software product containing program code for performing the method according to claim 19.

83. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 54.

84. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 55.

85. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 56.

86. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 57.

87. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 58.

88. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 59.

89. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 60.

90. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 61.

91. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 62.

92. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 63.

93. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 64.

94. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 65.

95. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 66.

96. A switching device for controlling communication of signals between a source and an output, wherein the switching device comprises the buffer management system of claim 67.